

MIGRATION **VERS** **LES** STANDALONE **COMPONENTS**

Modernisez votre architecture
Angular pour plus de
performance et de maintenabilité



Elodie TURLIER

Angular, IA, agilité &
storytelling : construire des apps
prêtes à survivre à l'apocalypse
digitale



POURQUOI MIGRER ?

40%

Réduction du boilerplate

30%

Amélioration des performances

60%

Réduction des fichiers



Elodie TURLIER

Angular, IA, agilité & storytelling



COMPARAISON ARCHITECTURALE

```
● ● ●  
  
// Ancienne approche avec NgModule  
@NgModule({  
  declarations: [LegacyComponent],  
  imports: [CommonModule]  
})  
export class LegacyModule {}  
  
// Nouvelle approche standalone  
@Component({  
  standalone: true,  
  imports: [CommonModule]  
})  
export class StandaloneComponent {}
```

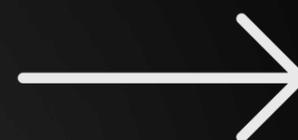
Plus besoin de spécifier *standalone: true*
car par défaut depuis Angular 19

Avantage clé : Les standalone components éliminent
les couches architecturales superflues tout en
conservant la compatibilité avec le code existant.



Elodie TURLIER

Angular, IA, agilité & storytelling



AUDIT PRÉLIMINAIRE

✔ **Vérifier Angular v15.2+**

Version minimum requise pour les schematics de migration

✔ **Analyser les dépendances**

Vérifier la compatibilité des bibliothèques tierces

✔ **Créer un point de restauration**

Branche à part ou commit Git de départ

✔ **Cartographier les modules critiques**

CoreModule, SharedModule, ... et leurs dépendances

✔ **Analyse de Bundle initiale**

ng build --stats-json

npx source-map-explorer dist/*.js



Elodie TURLIER

Angular, IA, agilité & storytelling



CHECKLIST DE PRÉPARATION

Mise à jour Angular CLI

Latest ou v15.2+

Liste des bibliothèques non compatibles et celles à upgrader

Configuration budgets de performance

Mise en place ESLint avec règles spécifiques

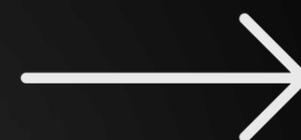
Exemple : détection d'imports superflus

Identifier les composants pilotes pour une migration initiale + ceux pour les différentes vagues de migrations



Elodie TURLIER

Angular, IA, agilité & storytelling



MIGRATION AUTOMATISÉE

Utilisation des **schematics** Angular pour une migration simplifiée: **Processus en 3 Étapes**

1. Conversion des composants :

```
ng generate @angular/core:standalone --mode=convert-to-standalone
```

2. Nettoyage des modules :

```
ng generate @angular/core:standalone --mode=prune-ng-modules
```

3. Bootstrap standalone :

```
ng generate @angular/core:standalone --mode=standalone-bootstrap
```



Elodie TURLIER

Angular, IA, agilité & storytelling



MIGRATION AUTOMATISÉE PAR VAGUES

Conversion des composants pilotes en utilisant le **schematic** officiel pour convertir des sous-ensembles fonctionnels :

```
ng generate @angular/core:standalone --  
mode=convert-to-standalone --path=src/app/shared
```

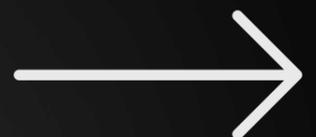
Exemple de modifications:

```
// Avant  
@Component({...})  
export class ExampleComponent {}  
  
// Après  
@Component({  
  standalone: true,  
  imports: [CommonModule, RouterLink]  
})  
export class ExampleComponent {}
```



Elodie TURLIER

Angular, IA, agilité & storytelling



GESTION DES DÉPENDANCES COMPLEXES

07

Pour les modules tiers non migrables : Utiliser *importProvidersFrom* pour l'interopérabilité

```
// Solution temporaire avec importProvidersFrom
import { provideRouter, importProvidersFrom } from '@angular/core';
import { LegacyLibModule } from 'legacy-lib';

bootstrapApplication(RootComponent, {
  providers: [
    importProvidersFrom(LegacyLibModule.forRoot()),
    provideHttpClient()
  ]
});
```

Cette approche permet une transition progressive vers les APIs standalone.



Elodie TURLIER

Angular, IA, agilité & storytelling



MIGRATION DES SERVICES

Services globaux :

Migrer progressivement vers *providedIn: 'root'*

Services partagés par plusieurs composants:

Utilisation du **providers** array au niveau des **routes**

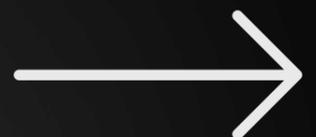
Services utilisé par un seul composant :

Utilisation du **providers** array au niveau du **composant**



Elodie TURLIER

Angular, IA, agilité & storytelling



OPTIMISATION DU LAZY LOADING

09

Adapter le routing pour exploiter les standalone components.

```
const routes: Routes = [  
  {  
    path: '',  
    pathMatch: 'full',  
    redirectTo: 'home'  
  },  
  {  
    path: 'home',  
    component: HomeComponent  
  },  
  // Option 1: Lazy Loading d'une autre config de routed  
  {  
    path: 'booking',  
    loadChildren: () =>  
      import('./booking/booking.routes')  
        .then(m => m.bookingRoutes)  
  },  
  // Option 2: Lazy Loading d'un standalone component  
  {  
    path: 'flight',  
    loadChildren: () =>  
      import('./flight/flight.component').then(m =>  
m.FlightComponent)  
    providers: [AdminService] // Providers scopés à la route  
  },  
  ...  
];  
  
bootstrapApplication(App, {  
  providers: [provideRouter(routes)],  
});
```



Elodie TURLIER

Angular, IA, agilité & storytelling



ADAPTATION DES TESTS

✓ **Mettre à jour les fichiers specs** pour refléter les imports standalone.

✓ Utiliser **TestBed.configureTestingModule** avec les nouveaux schémas

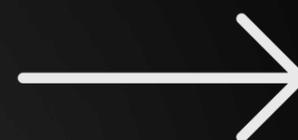
```
TestBed.configureTestingModule({
  imports: [StandaloneComponent]
});
```

✓ **Mettre à jour les tests E2E** pour valider les scénarios critiques



Elodie TURLIER

Angular, IA, agilité & storytelling



FINALISATION POST- MIGRATION

Suppression des NgModules

Exécuter le schematic de nettoyage après chaque vague:

```
ng generate @angular/core:standalone --mode=prune-ng-modules
```

Adaptation manuelle des Directives et Pipes non couverts par les schematics et adapter la même stratégie que pour les services

Migrer le point d'entrée principal avec **bootstrapApplication** et **suppression définitive de l'AppModule**

Utiliser des outils avancés pour valider les gains :

```
ng build --stats-json
```

```
npx source-map-explorer dist/*.js
```



Elodie TURLIER

Angular, IA, agilité & storytelling



RISQUES COURANTS ET SOLUTIONS

10

Symptôme	Cause	Correction
<code>NullInjectorError</code>	Providers manquants	Ajouter <code>provideHttpClient()</code>
<code>Component not found</code>	Import manquant	Vérifier les <code>imports</code> parents
<code>Unexpected directive</code>	Directive non migrée	Convertir en standalone

Outils de Débogage

- *Angular DevTools* - Analyse de l'arbre de dépendances
- *Source Map Explorer* - Visualisation des bundles
- *ESLint Rules* - Détection des anti-patterns
- *ng serve --verbose* - Logging détaillé

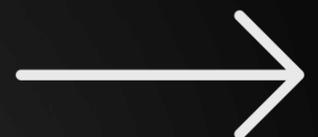
Pour plus d'info, la doc officielle:

<https://angular.dev/reference/migrations/standalone>

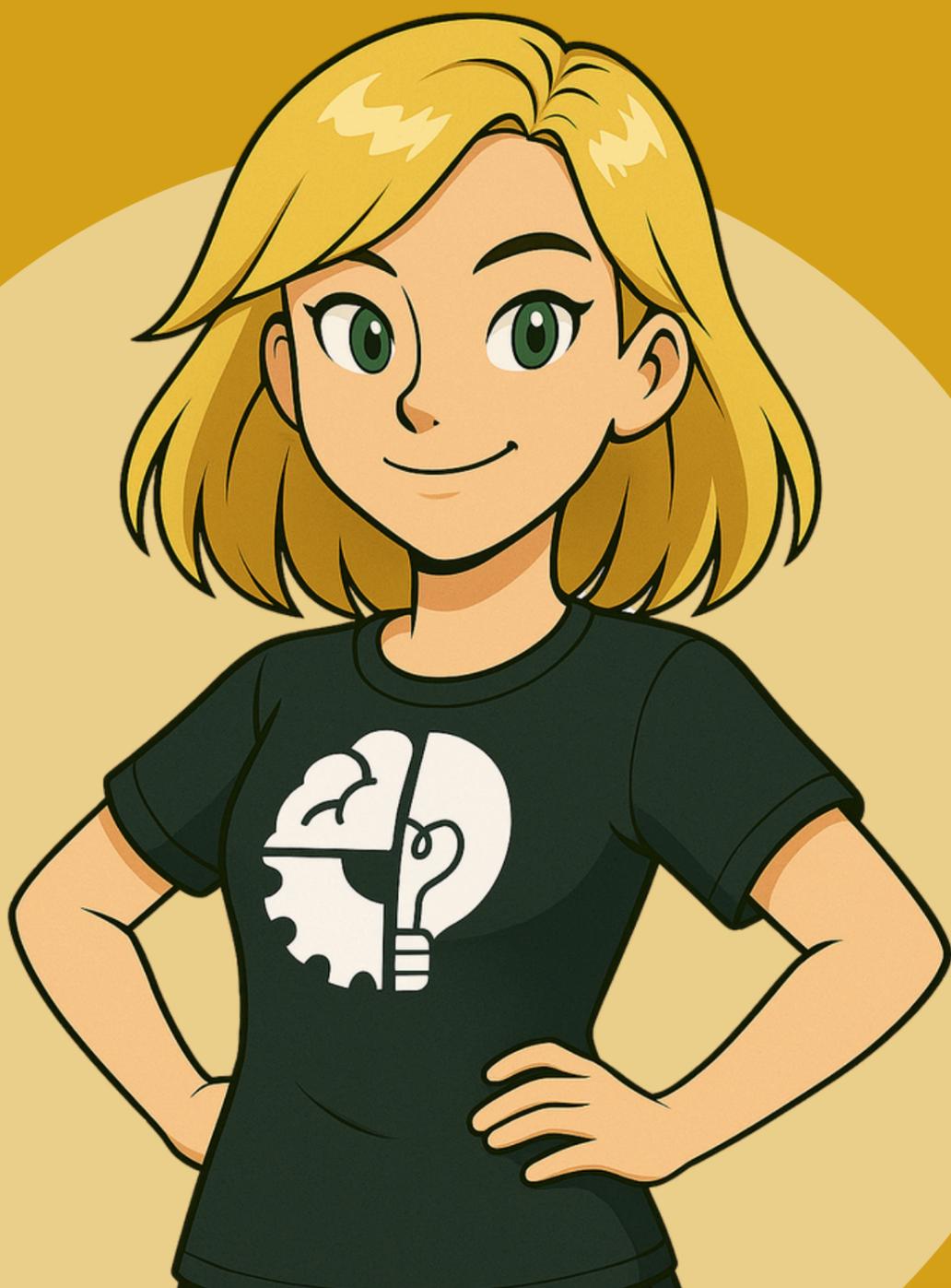


Elodie TURLIER

Angular, IA, agilité & storytelling



ABONNE TOI POUR PLUS DE CONTENU



Elodie TURLIER

Angular, IA, agilité & storytelling : construire des apps prêtes à survivre à l'apocalypse digitale



Enregistre ou partage si ça t'a été utile